PL-TR-94-2278

DATA RE-FORMATTER FOR THE TOPSIDE THERMAL PLASMA MONITOR (PLIESPP)

Andrew J. Mazzella, Jr.
J. Robert Cornelius

RDP Incorporated
391 Totten Pond Road
Waltham, Massachusetts 02154

20 October 1994

Scientific Report No. 5

Approved for public release; distribution unlimited

PHILLIPS LABORATORY
Directorate of Geophysics
AIR FORCE MATERIEL COMMAND
HANSCOM AIR FORCE BASE, MA 01731-3010

"This technical report has been reviewed and is approved for publication"

EDWARD C. ROBINSON
Contract Manager
Data Analysis Division

ROBERT E. McINERNEY, Director
Data Analysis Division

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/IM, 29 Randolph Road, Hanscom AFB, MA 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 20 October 1994	3. REPORT TYPE AND DATES COVERED Scientific Report No. 5

4. TITLE AND SUBTITLE Data Re-Formatter Program for the Topside Thermal Plasma Monitor (PLIESPP)	5. FUNDING NUMBERS PE 35160F PR 7659 TA 05 WU AC Contract: F19628-89-C-0079
6. AUTHOR(S) Andrew J. Mazzella, Jr. J. Robert Cornelius	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) RDP Incorporated 391 Totten Pond Road Waltham, Massachusetts 02154	8. PERFORMING ORGANIZATION REPORT NUMBER RDP-TR-9403
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Phillips Laboratory 29 Randolph Road Hanscom Air Force Base, Massachusetts 01731-3010 Contract Manager: Edward C. Robinson/GPD	10. SPONSORING/MONITORING AGENCY REPORT NUMBER PL-TR-94-2278
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The data re-formatting program (PLIESPP) for the SSIES thermal plasma monitor allows data from the Phillips Laboratory, Geophysics Directorate (PL/GP) Time History Data Base (THDB) format to be used by the Space Forecast Center (SFC) analysis program (APGA). This version of the PLIESPP program was designed to process data for the SSIES-2 telemetry formats, with provisions for adaptation to processing the older SSIES format or the SSIES-3 format, planned for future instruments.

14. SUBJECT TERMS Ionosphere, Plasma analysis, Particle detectors, DMSP, SSIES	15. NUMBER OF PAGES 62
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7504-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18 298-102

Table of Contents

1 Data Re-Formatter Program for the Topside Thermal Plasma Monitor (PLIESPP).

PLIESPP reads, unpacks, verifies, and reformats SSIES data in the Phillips Laboratory (PL) Time History Data Base (THDB) format for processing by APGA (IESCOMIN), the IES processing program. The input data file, created at PL on a CYBER or CONVEX processing system, consists of binary data records with the following characteristics:

1. The data are received from AFGWC, and transcribed at PL, in the Sperry/UNIVAC 36-bit word, 9-bit byte format.
2. The one-minute Ephemeris and Telemetry data records are converted to time sequenced (time ascending) order from the original time-reversed order.
3. The 60 seconds of data within each Telemetry record are in time sequenced order.
4. The Telemetry items are stored in the original UNIVAC byte sequence, which is the same convention used on the VAX but reversed from that used on the CYBER or CONVEX.
5. The size and format of the Telemetry data records are not consistent among the existing and proposed instrument configurations, e.g.: SSIES, SSIES-2, SSIES-2A, and SSIES-3, so padding occurs in the THDB.

1.1 Sub-Program Structure.

This section describes the modular hierarchy for the PLIESPP program, with a brief description of each routine. For a normal error-free run, the routines are invoked in the order presented.

PLIESPP	Main Program
. USERIN:	Request run parameter inputs Subroutine
.. CHEK_IN:	Verify run parameter inputs Subroutine
. RD3MIN:	Read 3-minute block from THDB Subroutine
. EPHEXT:	Read and unpack Ephemeris data Subroutine
.. IBITS:	VAX Fortran bit manipulation Function
.. BTEST:	VAX Fortran bit test Function
.. TIME_CHEK:	Verify date and time values Subroutine
. MAKERIR:	Create Readout Information Record Subroutine
.. CALNDR:	Convert Julian date to calendar date Subroutine
. EPHPROC:	Calculate ephemeris values Subroutine
.. PHASANG:	Compute orbital phase angle Subroutine
... XPROD:	Compute vector cross product Subroutine
... VMAG:	Compute vector magnitude Function
... DOTPRD:	Compute vector scalar product Function
. TMEXT:	Read and unpack Telemetry data Subroutine
.. IBITS:	VAX Fortran bit manipulation Function
.. MVBITS:	VAX Fortran bit manipulation Subroutine
. GET_XCEPTS:	Reformat Telemetry data Subroutine
.. IES_XCEPT:	Process exceptions for IES data Subroutine
... VALCHEK:	Verify Telemetry data Subroutine

... CYCNT1:	Cycle 1 specific processing Subroutine
.... VALCHEK:	Verify Telemetry data Subroutine
.... CYCNT2:	Cycle 2 specific processing Subroutine
... CYCNT2:	Cycle 2 specific processing Subroutine
.... VALCHEK:	Verify Telemetry data Subroutine
.... CYCNT1:	Cycle 1 specific processing Subroutine
.. IES2_XCEPT:	Process exceptions for IES2 data Subroutine
... VALCHEK:	Verify Telemetry data Subroutine
... CYCNT1:	Cycle 1 specific processing Subroutine
.... VALCHEK:	Verify Telemetry data Subroutine
.... CYCNT2:	Cycle 2 specific processing Subroutine
... CYCNT2:	Cycle 2 specific processing Subroutine
.... VALCHEK:	Verify Telemetry data Subroutine
.... CYCNT1:	Cycle 1 specific processing Subroutine
.. IES2A_XCEPT:	Process exceptions for IES2A data Subroutine
... VALCHEK:	Verify Telemetry data Subroutine
... CYCNT1:	Cycle 1 specific processing Subroutine
.... VALCHEK:	Verify Telemetry data Subroutine
.... CYCNT2:	Cycle 2 specific processing Subroutine
... CYCNT2:	Cycle 2 specific processing Subroutine
.... VALCHEK:	Verify Telemetry data Subroutine
.... CYCNT1:	Cycle 1 specific processing Subroutine
.. IES3_XCEPT:	Process exceptions for IES3 data Subroutine (This is currently not implemented.)
. STOREM:	Store reformatted Telemetry block Subroutine
. OUTPUT:	Write reformatted RIR/Ephemeris/Telemetry data blocks Subroutine

1.2 Inputs.

The PLIESPP program requires one input data file and a series of user-supplied interactive inputs.

1.2.1 Input Data File.

The data reformatter program (PLIESPP) uses a single input file. This unformatted binary input file is created in a sequential form by the PHASE I generating program. The contents and formats of the file are described in the technical report Database Development for the DMSP Experiments (GL-TR-90-0028), with a slight modification for SSIES-2 (see Appendix A of this document). The file consists of a sequence of 3-minute physical records, with each minute containing a logical Ephemeris record and 60 logical 1-second Telemetry records. A logical trailer record follows the three minutes of data.

The Telemetry record for a given minute of orbit follows the Ephemeris record for the same time period. Each Telemetry record consists of 60 sets of packed Telemetry data, with each set representing one second of time within the minute. Missing seconds of data do not appear interspersed within the time sequence, but are compensated for in the record length by zero-filled records at the end of the 1-minute interval, and are indicated by zero bits in an index word included with the Ephemeris record.

1.2.2 User Inputs.

User inputs are supplied either interactively, in response to prompts, at the user console, or in a run file for a batch process. The required inputs are as follows:

1. The INPUT file name with full directory path (up to 48 characters).
2. The OUTPUT file name with full directory path (up to 48 characters).
3. The code for the instrument type being processed, i.e.: 1 for SSIES; 2 for SSIES-2; 3 for SSIES-2A; 4 for SSIES-3
4. The 2-digit Flight (Satellite) Identifier.
5. The data Start time of interest as YY DDD HH MM SS
6. The data End time of interest as YY DDD HH MM SS

   Where:
   - YY is the 2 digit Year, i.e.: 93 for 1993.
   - DDD is the Julian Day of the Year.
   - HH is the Hour of the day.
   - MM is the Minute of the Hour.
   - SS is the Second of the Minute.

   Note: The start time must always be less than the end time because the input file data are in time ascending order.

   The input values are checked, and, if found to be out of range, the erroneous value is displayed, and the user is prompted to enter correct values.

7. The program will echo the inputs selected and request the user to input YES to proceed or NO to modify the input values.

1.3 Outputs.

This section describes the files and informative messages generated during the execution of the PLIESPP program.

1.3.1 Output Data Files.

The IESPREPFILE, the primary output of the PLIESPP program, is a sequential, unformatted binary file with a fixed record size of 4888 words. The first record contains a reconstructed 10-word RIR data record. The second and successive records contain a 28-word Ephemeris data block, a 60-word data frame time block, and a 4800-word Telemetry data frame block. The data frame block consists of 60 data frames, each containing 160 VAX 16-bit half-words.

The IESPREPFILE is subsequently used as the primary input to the SSIES processing program (APGA/IESCOMIN). The exact format of the records contained in this file appears in Appendix B.

An optional output print file (FOR080.DAT) may be obtained by activating the Subroutine PRINTIT. This ASCII file contains the unpacked Ephemeris data and the unpacked raw Telemetry data before reformatting. This print option should be used over short time spans of data for debug or verification purposes only.

1.3.2 Messages.

Several self-explanatory prompts which appear during user input processing are described in section 1.2.2. In addition to these user prompts, PLIESPP generates the following informative message after the first Ephemeris record has been processed:

PROCESSING SATELLITE nnn
WITH GMF VERSION ID OF aaaaaa
FOR DAY ddd OF 19yy

nnn	is the flight number
aaaaaaa	is the GeoMagnetic Field model identifier
ddd	is the day of year for the data
yy	is the decade portion of the year for the data.

1.4 Error recovery and handling.

Informative error messages are printed to alert the user of problems encountered in processing the raw data stream. These messages are generally followed by program termination. The messages contain enough information for the user to determine the cause of the error and correct it before attempting another run.

REQUESTED INPUT FILE "input file path and name"
CONTAINS DATA FOR SATELLITE nn
ON DAY ddd OF YEAR 19yy
YOU REQUESTED DATA FOR SATELLITE mm
ON DAY jjj OF YEAR 19rr
JOB TERMINATED.

The specified input file contains data which do not correspond to the requested Satellite (nn) or day (ddd) of the designated year (19yy).

The run will be terminated; no output file will be written.

1.5 Processing Time.

The CPU time required for PLIESPP to process one orbit (~101 minutes) of data is about 20 seconds on a VAX 7000.

2 Sample Inputs and Outputs.

a. A sample script of user prompts and replies for PLIESPP (user responses are in italics, with explanatory comments following the "/").

$ R PLIESPP
Enter the INPUT File Name (with path if necessary)
F11B25OCT93.DAT */input file name: TmFileNam*
Enter the OUTPUT File Name (with path if necessary)
F11B25OCT93.PRP */output file name: IESPrepFil*
Enter Instrument Type: 1 for SSIES, 2 for SSIES-2
3 for SSIES-2A, 4 for SSIES-3
2 */instrument type (SSIES-2A): IESReq*
Enter Satellite Id as a 2 digit INTEGER.
11 */satellite ID: NumSat*
Enter Start Time as YY DDD HH MM SS
93 298 08 00 00 */Start time: InYr InDay InHr InMin InSec*
Enter End Time as YY DDD HH MM SS
93 298 10 00 00 */Start time: NdYr NdDay NdHr NdMin NdSec*
We will process file F11B25OCT93.DAT
for SSIES-2 Satellite Id 11
from 93 298 8 0 0 28800
to 93 298 10 0 0 36000
and write the results to F11B25OCT93.PRP

Type YES to proceed.
Type NO to change specifications.
YES */Confirmation for specifications*

3 Data File Descriptions

3.1 Data File Format for Input to PLIESPP at Phillips Laboratory.

The input data file for PLIESPP is an unformatted binary file with a sequential file structure. The fixed-length records (19080 bytes, or 4770 VAX 32-bit words) of the file are accessed using successive binary READs.

There is only one type of data record represented in the file. Each data record contains three pairs of a 1-minute Ephemeris data header followed by a 1-minute block of 60 telemetry data frames, all followed by a three-item trailer. The Ephemeris header is 420 bits in length, while each minute of telemetry data is 50400 bits (6300 bytes) in length. The trailer items total 180 bits in length.

Each telemetry data frame contains 1 second of data. The data frame times correspond to the time span designated in the preceding Ephemeris data record. The exact content and format of the record is described in Database Development for the DMSP Experiments, Appendix B, and in Appendix A of this document.

3.2 Data File Format Output by PLIESPP for APGA.

The IESPREPFILE is the principal output of the PLIESPP program and the principal input file of the APGA program. The IESPREPFILE is an unformatted binary file with fixed length 4888-word records. The first record contains the 10-word Readout Information Record (RIR). The second and successive records contain a 28-word Ephemeris data block, a 60-word data frame

time block, and a 4800-word instrument data frame block. The data frame block contains 60 data frames, each frame consisting of 160 16-bit half-words of instrument data. The format and content of each of these data blocks are listed in the Appendix B.

4 Program PLIESPP Description.

4.1 Main Routine.

a. Function.

Control the unpacking, reordering, and reformatting of SSIES, SSIES2, and SSIES3 Phase I data streams to produce a Common Format Prepfile for the APGA (IESCOMIN) processing program.

b. Inputs.

1. Files:

TMFILE - File containing the Phase I format data, with a restructured ephemeris block and packed SSIES, SSIES2, or SSIES3 telemetry data.

c. Processing.

1. Invoke Subroutine USERIN to obtain run control parameters from the user.

2. Initialize data storage arrays.

3. Invoke Subroutine RD3MIN to open the input file and resequence the data record for unpacking.

4. Invoke Subroutine EPHEXT to extract the ephemeris information for one minute of data.

5. Verify that the input file satellite number and date match the requested satellite number and date.

6. Invoke Subroutine MAKERIR to construct a Readout Information Record from the ephemeris and user-supplied information.

7. Process the first minute of data, creating a properly formatted ephemeris record and unpacking the telemetry data.

8. Establish the unpacking loop to:

   a. read the Ephemeris and Telemetry for the next minute,

   b. unpack and process the Ephemeris data record for the minute,

   c. unpack a telemetry data record for the minute,

   d. invoke Subroutine GET_XCEPTS to process the Common Format storage exceptions for the minute,

e. write the unpacked data for the previous minute to the PrepFile,

f. loop through step 8 until all requested data have been unpacked.

9. Because the data reformatting involves the transfer of data from one cycle (record) to another, the input data are double buffered.

d. Outputs.

1. Files.

IESPREPFIL - The unpacked, reordered, and reformatted Ephemeris and telemetry data. This file is used as the primary input file to APGA.

e. Associated subroutines.

1. Subroutine USERIN. Acquire user specified processing control parameters.

2. Subroutine RD3MIN. Open the input file and read a three-minute data record.

3. Subroutine EPHEXT. Read and unpack an Ephemeris data record.

4. Subroutine MAKERIR. Generate a Readout Information Record.

5. Subroutine EPHPROC. Convert Ephemeris data to the proper units and format.

6. Subroutine TMEXT. Read and unpack a telemetry data record.

7. Subroutine GET_XCEPTS. Process the Common Format storage exceptions.

8. Subroutine STOREM. Store the telemetry data in the Common Format.

9. Subroutine OUTPUT. Write a reformatted data record to the output file.

f. Interfaces.

1. Main routine.

2. Uses COMMON blocks EXTRACT, OUTREC, and WORKREC.

4.1.1 Subroutine USERIN.

a. Function.

Prompt the user for necessary control parameters, and verify the validity of the values entered.

b. Inputs.

All inputs are parameters supplied by the user.

c. Processing.

1. Prompt the user to supply the input file path and name.

2. Prompt the user to supply the output file path and name.

3. Prompt the user to supply the instrument type, and validity check the response (IES, IES2, IES2A, IES3).

4. Prompt the user to supply a 2-digit satellite ID.

5. Prompt the user to supply a start time (Year, Day of year, Hour, Minute, Second) for the processing, and validity check the response.

6. Prompt the user to supply an end time (Year, Day of year, Hour, Minute, Second) for the processing, and validity check the response.

7. Echo the user supplied parameters and prompt the user for a response to proceed or modify the parameters.

8. Return to program PLIESPP.

d. Outputs.

1. Argument list.

TMFILENAM	-	Full path name of the input file
IESPREPFIL	-	Full path name of the output file
IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
NUMSAT	-	Specified satellite ID
INYR	-	Year for initial selection of data, in 2-digit form
INDAY	-	Day of year for initial selection of data, in 3-digit form
INTIME	-	Time to start processing (seconds)
INDATE	-	Combined initial year and day of year
NDYR	-	Year for ending selection of data, in 2-digit form
NDDAY	-	Day of year for ending selection of data, in 3-digit form
NDTIME	-	Time to end processing (seconds)
NDDATE	-	Combined ending year and day of year
PROCEED	-	Proceed with processing flag (LOGICAL)

e. Associated subroutines.

1. Subroutine CHEK_IN. Check validity of user supplied times, and convert Hour, Minute, Second to total seconds.

f. Interfaces.

1. Called by the main routine.

2. Uses no COMMON blocks.

4.1.1.1 Subroutine CHEK_IN.

a. Function.

Validate the user-supplied date and time values for data selection.

b. Inputs.

1. Argument list.

KYR	-	Two-digit year designation
KDAY	-	Day of year
KHR	-	Hour of day
KMIN	-	Minute of hour
KSEC	-	Second of minute

2. COMMON blocks.

None.

c. Processing.

1. Compare year to allowed range and set error flags if out of range.

2. Compare day to allowed range and set error flags if out of range.

3. Compare hour to allowed range and set error flags if out of range.

4. Compare minute to allowed range and set error flags if out of range.

5. Compare second to allowed range and set error flags if out of range.

6. Check status of error flags:

   a. If set for error, report invalid value.

   b. If clear of error, set combined year and day, and time in seconds.

7. Return to USERIN.

d. Outputs.

1. Argument list.

KTIME	-	Time of day in seconds
KDATE	-	Combined year and day of year (as YYDDD)

e. Associated subroutines.

None.

f. Interfaces.

1. Called by USERIN.

2. Uses no COMMON blocks.

4.1.2 Subroutine RD3MIN.

a. Function.

Acquire a physical record containing three minutes of data.

b. Inputs.

1. Argument list.

TMFILENAM	-	Input file name
INUNIT	-	Input file unit number

2. COMMON blocks.

/EXTRACT/

IESTYP	-	Instrument type identifier

3. Files.

TMFILE - The raw data file from which the data records are read.

c. Processing.

1. For the first invocation only, open the designated (TMFILENAM) input data file. If the open was not successful,

    a. Set the proper indicators,

    b. Return to PLIESPP.

2. Acquire data, using the physical and logical record lengths appropriate for the satellite type.

3. Reorder the data for subsequent unpacking.

4. Store the Geomagnetic Field Model identifier.

5. Return to PLIESPP.

d. Outputs

1. Arguments

IDSAT	-	Satellite ID number
GMFLBL	-	Geomagnetic Field model identifier
EOF	-	Flag for end-of-file condition

2. COMMON blocks

/EXTRACT/

IPAK3MIN	-	Resequenced, packed Ephemeris and Telemetry data

e. Associated subroutines.

None.

f. Interfaces

1. Called from main routine.

2. Uses COMMON block EXTRACT.

4.1.3 Subroutine EPHEXT.

a. Function.

Unpack, convert, and store the Ephemeris data for use by the APGA processing program.

b. Inputs.

1. Argument list.

ICUR	-	Pointer to current data record
IPREV	-	Pointer to previous data record

2. COMMON blocks.

/EXTRACT/

IESTYP	-	Instrument type identifier
MINIT	-	Selected minute within three-minute block
IPAK3MIN	-	Packed Ephemeris and Telemetry data

c. Processing.

1. Initialize the word (LWORD) and start-bit within the word (INBIT) pointers to begin the unpacking process at the end of the bit stream.

2. Establish a loop to unpack each of the 27 data quantities available, using the VAX

intrinsic function IBITS. The size of each quantity (in bits) is specified in the array NBITS, and the word and start bit within the word pointers are decremented as needed.

a. For the INTEGER quantities being unpacked, the appropriate number of bits are extracted.

b. Where required, the most significant bit is checked as a sign bit, and the complement operation is performed.

c. The converted value is stored into the current output array.

3. Extract the mapping bits for the proper sequence of available and missing data seconds.

4. Set and verify (using TIME_CHEK) the dates and times for the Ephemeris block.

5. Return to PLIESPP.

d. Outputs.

1. Argument list.

ICUR	-	Pointer for current minute
IPREV	-	Pointer for previous minute
IEPHUPK	-	Two sequential minutes of unpacked Ephemeris values
IOATIME	-	Two date and time pairs, for the associated Ephemeris values
MAPSEC	-	Array for index mapping of the Telemetry data

2. Common blocks.

/EXTRACT/

LWORD	-	Word pointer for current unpacking location
IBIT	-	Bit pointer for current unpacking location

e. Associated subroutines.

1. Subroutine TIME_CHEK. Date and time verification.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block EXTRACT.

4.1.3.1 Subroutine TIME_CHEK.

a. Function.

Verify date and time values.

b. Inputs.

1. Argument list.

IDATE	-	Combined year and day-of-year
ITIME	-	Time of day in seconds

c. Processing.

1. Compare the day and time against the allowable ranges.

   a. If the values are acceptable, adjust the day and time to accommodate any day rollovers for the given time value,

   b. Adjust the year and day-of-year to accommodate any year rollovers,

   c. Calculate the resulting combined year and day-of-year value.

2. If the date value is not acceptable, both of the returned values are set to zero.

d. Outputs.

1. Argument list.

IDATE	-	Combined year and day-of-year
ITIME	-	Time of day in seconds

e. Associated subroutines.

None.

f. Interfaces.

1. Called by EPHEXT.

2. Uses no COMMON blocks.

4.1.4 Subroutine MAKERIR.

a. Function.

Create a Readout Information Record from the ephemeris and user-supplied information.

b. Inputs.

1. Argument list.

IESTYP	-	Instrument type identifier
NUMSAT	-	Specified satellite ID
IDDAY	-	Day of year for selection of data
IDYR	-	Year for selection of data, in 2-digit form

INTIME	-	Time to start processing (seconds)
NDTIME	-	Time to end processing (seconds)

c. Processing.

1. Invoke CALNDR to determine the month and day-of-month from the year and day-of-year.

2. Utilize the calendar date, input items, and internal Mission ID table to define the RIR items.

3. Return to PLIESPP.

d. Outputs.

1. COMMON blocks.

/OUTRIR/

IRIROUT	-	Array of RIR data

e. Associated subroutines.

1. Subroutine CALNDR. Determine the month and day-of-month from the year and day-of-year.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block OUTRIR.

4.1.4.1 Subroutine CALNDR

a. Function.

Determine the month and day-of-month from the year and day-of-year.

b. Inputs.

1. Argument list.

IYEAR	-	Year of date
JDAY	-	Day-of-year

c. Processing.

1. Determine whether the designated year is a leap year.

   a. Set the cumulative day offset to one if it is a leap year.

b. Leave the cumulative day offset as zero if it is not a leap year.

2. For each month, compare the designated day-of-year against the cumulative days in the year, including that month.

   a. When the designated day is less than the cumulative days, assign the value for the month,

   b. Assign the residual over cumulative days from the previous month as the value for the day-of-month.

3. Return to MAKERIR.

d. Outputs.

1. Argument list.

MON	-	Numerical month value for date
MDAY	-	Day-of-month value

e. Associated subroutines.

None.

f. Interfaces.

1. Called by MAKERIR.

2. Uses no COMMON blocks.

4.1.5 Subroutine EPHPROC.

a. Function.

Calculate Ephemeris values from header values in THDB.

b. Inputs.

1. Argument list.

IEPHUPK	-	Two sequential minutes of unpacked Ephemeris values

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current data record
IPREV	-	Pointer to previous data record

c. Processing.

1. Map the available Ephemeris values from the THDB header into the required words of the Ephemeris record, combining values, as necessary, and converting units.

2. Invoke PHASANG to calculate the orbital phase angles, using the locations reported for two individual minutes.

3. Store the orbital phase angles in the Ephemeris record.

4. Set unused Ephemeris record values to zero.

5. Return to PLIESPP.

d. Outputs.

1. COMMON blocks.

/WORKREC/

IPHVAL	-	Unpacked Ephemeris data buffer

e. Associated subroutines.

1. Subroutine PHASANG. Calculate the orbital phase angle.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block WORKREC.

4.1.5.1 Subroutine PHASANG.

a. Function.

Calculate the orbital phase angle, in radians.

b. Inputs.

1. Argument list.

EPHVAL	-	Ephemeris record array

c. Processing.

1. Calculate the Cartesian coordinates for the two locations stored in the Ephemeris record.

2. Use the two locations to determine the axis of the orbit.

3. Use the orbit axis and the earth's axis to determine the location of the ascending node.

4. Calculate the azimuthal angle about the orbit axis from the ascending node to each of the two locations in the Ephemeris record.

5. Return to EPHPROC.

d. Outputs.

1. Argument list.

ORBANG	-	Orbital phase angles for the two locations.

e. Associated subroutines.

1. Subroutine XPROD. Calculate the vector cross-product of two vectors.

2. Function VMAG. Calculate the magnitude of a vector.

3. Function DOTPRD. Calculate the dot (scalar) product of two vectors.

f. Interfaces.

1. Called by EPHPROC.

2. Uses no COMMON blocks.

4.1.5.1.1 Subroutine XPROD.

a. Function.

Calculate the vector cross-product of two vectors.

b. Inputs.

1. Argument list.

A	-	First vector of product
B	-	Second vector of product

c. Processing.

1. Calculate each component of the resultant vector, using the appropriate components of the input vectors.

2. Return to PHASANG.

d. Outputs.

1. Argument list.

C	-	Components of vector product.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by PHASANG.

2. Uses no COMMON blocks.

4.1.5.1.2 Subroutine VMAG.

a. Function.

Calculate the magnitude of a vector.

b. Inputs.

1. Argument list.

V - Components of vector.

c. Processing.

1. Calculate the square-root of the sum of the squares of the vector components.

2. Return to PHASANG.

d. Outputs.

1. Function value.

VMAG - Magnitude of vector.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by PHASANG.

2. Uses no COMMON blocks.

4.1.5.1.3 Subroutine DOTPRD.

a. Function.

Calculate the dot (scalar) product of two vectors.

b. Inputs.

1. Argument list.

A	-	First vector of product
B	-	Second vector of product

c. Processing.

1. Compute the sum of the products of the input vector components.

2. Return the result to PHASANG.

d. Outputs.

1. Function value.

DOTPRD	-	Scalar product

e. Associated subroutines.

None.

f. Interfaces.

1. Called by PHASANG.

2. Uses no COMMON blocks.

4.1.6 Subroutine TMEXT.

a. Function.

Unpack, convert, and store the 60 sets of 1-second Telemetry data for use by the APGA processing program.

b. Inputs.

1. Argument list.

MAPSEC	-	Array of mapping bits, indicating valid or missing seconds

2. COMMON blocks.

/EXTRACT/

IESTYP	-	Instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
LWORD	-	Word pointer for current unpacking location
IBIT	-	Bit pointer for current unpacking location
IPAK3MIN	-	Packed Ephemeris and Telemetry data

c. Processing.

1. Initialize unpacking at the location designated by the word (LWORD) and start-bit within the word (INBIT) pointers, as set by the Ephemeris extraction.

2. Establish a loop to unpack each of the 60 sets of Telemetry data quantities needed, using the VAX intrinsic function IBITS. The size of each quantity (in bits) is specified by ITMSIZ, and the pointers to the word and to the start bit within the word are decremented as needed.

    a. Establish a loop to unpack the (instrument-dependent) number of sets of data each second. Note that a set is defined as 4 9-bit quantities, which equals 1 Sperry 36-bit word.

    b. The first set of data for each second contains the coded time for the record. These values are extracted and concatenated into a VAX longword.

    c. The remaining sets each contain 4 9-bit coded instrument data or control information. Each of these 9-bit values is extracted, aligned, and stored into the current output (VAX INTEGER * 2) array.

3. Arrange the Telemetry data into the proper time sequence, with interspersed zero-filled values for missing seconds.

4. The time value is converted into seconds and stored into the current time array.

5. When the data for all 60 seconds have been processed, return to PLIESPP.

d. Outputs.

1. COMMON blocks.

/WORKREC/

ITMUPK - Array of unpacked telemetry data

e. Associated subroutines.

1. Function IBITS. VMS bit manipulation function.

2. Subroutine MVBITS. VMS bit manipulation subroutine.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON blocks WORKREC, EXTRACT.

4.1.7 Subroutine PRINTIT.

a. Function.

Generate a diagnostic listing of the data.

Note: PRINTIT is activated only by editing the source code to remove the leading COMMENT designation, and is de-activated by restoring the COMMENT designation.

b. Inputs.

1. Argument list.

IESTYP	-	Instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
KOUT	-	Index for previous or current minute in Ephemeris array

2. COMMON blocks.

/WORKREC/

IPHVAL	-	Unpacked Ephemeris data buffer
ITMSEC	-	Telemetry data times buffer
ITMUPK	-	Unpacked telemetry data buffer

c. Processing.

1. Print the Ephemeris record data for the selected minute.

2. Print the unpacked Telemetry record for each second of the minute.

3. Return to PLIESPP.

d. Outputs.

1. Files.

FOR080.DAT - Text listing of data values.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block WORKREC.

4.1.8 Subroutine GET_XCEPTS.

Subroutine GET_XCEPTS and its supporting routines are common to both BNBA (SFCIESPP) and PLIESPP.

a. Function.

Control the processing of telemetry data exceptions for all instrument types. An exception is defined as:

1. A data quantity whose Common Format data cycle differs from its instrument telemetry data cycle.

2. A quantity whose Common Format data rate differs from its instrument telemetry data rate.

3. A data quantity which is created by PLIESPP from instrument telemetry data and stored in the Common Format.

b. Inputs.

1. Argument list.

IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1-minute data record
IPREV	-	Pointer to previous 1-minute data record
IXVAL	-	Array of telemetry storage exception values

3. Files.

None.

c. Processing.

1. Initialize the exceptions array for the current minute of data to filler values (-1).

2. Propagate any forward-filled data exceptions by copying the extra 4 cycles (61 - 64) from the previous minute's exception set into the first 4 cycles of the current minute's exception set.

3. Invoke the subroutine which processes exceptions for the specified instrument's telemetry data.

   a. When IESTYP is 1, invoke IES_XCEPT.
   b. When IESTYP is 2, invoke IES2_XCEPT.
   c. When IESTYP is 3, invoke IES2A_XCEPT.

d. When IESTYP is 4, invoke IES3_XCEPT.

4. Return to PLIESPP.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/WORKREC/

IXVAL	-	Array of telemetry storage exception values

e. Associated subroutines.

1. Subroutine IES_XCEPT. Process exceptions for the IES instrument telemetry data.

2. Subroutine IES2_XCEPT. Process exceptions for the IES2 instrument telemetry data.

3. Subroutine IES2A_XCEPT. Process exceptions for the IES2A instrument telemetry data.

4. Subroutine IES3_XCEPT. Process exceptions for the IES3 instrument telemetry data.

f. Interfaces.

1. Called by PLIESPP.

2. Uses COMMON block WORKREC.

4.1.8.1 Subroutine IES_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES instrument configuration. The specific IES exceptions are listed in Table 1.

b. Inputs.

1. Argument list.

None.

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1-minute data buffers
IPREV	-	Pointer to previous 1-minute data buffers

Table 1 IES exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
1	1	14	2	111/all	Ion Velocity
2	1	145	1	1/9	Cycle1 MSB
3	1	146	1	1/6-8	EP Mode
4	1	147	always=2 for IES		EP/RPA Flag
5	1	148	1	1/5	Test Mode
6	1	149	1	1/4	Bias Mode
7	1	150	1	1/3	Sweep Clock
8	1	151	1	1/2	PRF/Reset
9	1	155	1	90/2-3	VIP Setting
10	1	156	1	90/1-5	VBIAS Volts
11	1	158	1&2	1/	Cycle Count
1	2	15	1	61/all	DM Signal Level
2	2	145	1	1/9	Cycle1 MSB
3	2	146	1	1/6-8	EP Mode
4	1	147	always=2 for IES		EP/RPA Flag
5	2	151	2	70/9	VBias Monitor
6	2	155	2	90/7-8	VIP Setting
7	2	156	2	90/1-5	VBIAS Volts
8	2	158	1&2		Cycle Count

ITMUPK - Unpacked telemetry data buffers

3. Files.

None.

c. Processing.

1. Establish a loop to process each of the 60 one-second telemetry data blocks in the current 1-minute buffer.

2. Invoke Subroutine VALCHEK to validity check the specified second's worth of data.

   a. If the data are not valid, set the cycle indicator to the filler value (-1), set the break indicator to .TRUE. to indicate a discontinuity, and proceed to the next second's worth of data.

   b. If the data are valid, proceed to step 3.

3. Calculate the data cycle from the Cycle ID and store it in the data cycle indicator.

4. If this is a Cycle 1 data set:

   a. Invoke Subroutine CYCNT1 to extract the exceptions from the Cycle ID and the Configuration ID words.

   b. Save the exceptions which appear in the Cycle 1 instrument telemetry data set.

5. If this is a Cycle 2 data set:

   a. Invoke Subroutine CYCNT2 to extract the exceptions from the Cycle ID and Configuration ID words.

   b. Save the exceptions which appear in the Cycle 2 instrument telemetry data set.

6. When all 60 cycles (seconds) have been processed, return to Subroutine GET_XCEPTS.

d. Outputs.

1. Argument list.

   None.

2. COMMON blocks.

   /WORKREC/

NCYCLE	-	Array of Cycle indicators
IXVAL	-	Array of saved exceptions

e. Associated subroutines.

1. Subroutine VALCHEK. Validity check the telemetry data.

2. Subroutine CYCNT1. Process Cycle ID and Configuration ID for Cycle 1.

3. Subroutine CYCNT2. Process Cycle ID and Configuration ID for Cycle 2.

f. Interfaces.

1. Called by Subroutine GET_XCEPTS.

2. Uses COMMON block WORKREC.

4.1.8.1.1 Subroutine VALCHEK.

a. Function.

   Check the validity of the unpacked telemetry data for the specified second.

b. Inputs.

1. Argument list.

IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
J	-	Specified second of data

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1-minute data record
ITMUPK	-	Unpacked telemetry data buffer

3. Files.

None.

c. Processing.

1. Initialize the cycle indicator for this second of the current minute's telemetry data buffer.

2. When the telemetry Cycle ID is a non-zero positive number, the data are valid, unless the OLS/DSM command is not valid (greater than 159).

3. When the Cycle ID is zero or negative, the data are not valid when the following are all zero or negative:

   a. For the IES instrument:

      EP Sweep Voltage
      RPA Sweep Voltage
      SM (WIBAN1) Range
      DM Log Level Amplifier

   b. For IES2, IES2A, and (probably) IES3 instrument:

      Configuration ID
      DM Log Level Amplifier
      SM (WIBAN1) Range
      DM (WIBAN2) Range

   Otherwise, the data are considered valid.

4. When the data are not valid set this second's cycle indicator to a filler value (-1).

5. Return to the calling routine.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/WORKREC/

NCYCLE	-	Array of cycle indicators

e. Associated subroutines.

None.

f. Interfaces.

1. Called by IES_XCEPT, IES2_XCEPT, IES2A_XCEPT, IES3_XCEPT, CYCNT1, and CYCNT2.

2. Uses COMMON block WORKREC.

4.1.8.1.2 Subroutine CYCNT1.

a. Function.

Update the running cycle counter and extract the exceptions data from the Cycle 1 ID and Configuration ID words.

b. Inputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
J	-	Specified second of data
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1-minute data buffer
ITMUPK	-	Unpacked telemetry data buffer

3. Files.

None.

c. Processing.

1. Extract the MSB and the EP mode indicator bits from the Cycle ID and save the values in the exceptions array.

2. Adjust the EP mode indicator to the Common Format value when the data are from IES.

3. If this is the first cycle processed or the previous cycle contained invalid data (BREAK is .TRUE.):

    a. Set the running cycle counter to zero.

    b. If this is not the last second of data in the current buffer, invoke VALCHK to validity check the next second of data, which should be a Cycle 2 second.

    c. If the next second of data is valid, set the cycle interrupt flag (BREAK) to .FALSE. and invoke CYCNT2 to establish the running cycle counter.

4. Adjust the running cycle counter to the proper value for the (missing) previous cycle.

5. Extract the remaining bits of information from the Cycle ID word and store the values in the exceptions block.

6. When the instrument is IES2, IES2A, or IES3, extract the necessary bits from the Configuration ID word and store the values in the exceptions block.

7. Update the running cycle counter.

    a. If the running cycle counter is 1024, reset it to 1.

    b. If the EP mode is A, B, BS, or E and the running cycle counter is 128, reset it to 1.

    c. Otherwise increment the running cycle counter by 1.

8. Return to the calling routine.

d. Outputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

None.

e. Associated subroutines.

1. Subroutine VALCHEK. Validity check the telemetry data.

2. Subroutine CYCNT2. Process Cycle ID and Configuration ID for Cycle 2.

f. Interfaces.

1. Called by IES_XCEPT, IES2_XCEPT, IES2A_XCEPT, IES3_XCEPT, and CYCNT2.

2. Uses COMMON block WORKREC.

4.1.8.1.3 Subroutine CYCNT2.

a. Function.

Update the running cycle counter and extract the exceptions data from the Cycle 2 ID and Configuration ID words.

b. Inputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
J	-	Specified second of data
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1-minute data buffer
ITMUPK	-	Unpacked telemetry data buffer

3. Files.

None.

c. Processing.

1. If this is the first cycle processed or the previous cycle(s) contained invalid data (BREAK is .TRUE.):

   a. Set the running cycle counter to zero.

   b. If this is not the last second of data in the current buffer, invoke VALCHK to validity check the next second of data, which should be a Cycle 1 second.

   c. If the next second of data is valid, set the cycle interrupt flag (BREAK) to .FALSE. and invoke CYCNT1 to get the EP mode and the Cycle 1 ID MSB.

2. If this is IES instrument data, set the EP/RPA processing indicator to process both types of data.

3. If this is IES2, IES2A, or IES3 instrument data, extract the necessary bits from the Configuration ID and store the values for exceptions processing.

4. Calculate the running cycle counter from the Cycle 2 ID and the Cycle 1 MSB.

5. If the EP mode is C, D, or DS and:

   a. the running cycle counter is 0, reset it to 512,

   b. or, if the running cycle counter is 512, reset it to 1024,

   c. otherwise, if the running cycle counter is 0 reset it to 128.

6. Return to the calling routine.

d. Outputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values

NCYCNT - Running cycle counter

2. COMMON blocks.

None.

e. Associated subroutines.

1. Subroutine VALCHEK. Validity check the telemetry data.

2. Subroutine CYCNT1. Process Cycle ID and Configuration ID for Cycle 1.

f. Interfaces.

1. Called by IES_XCEPT, IES2_XCEPT, IES2A_XCEPT, IES3_XCEPT, and CYCNT1.

2. Uses COMMON block WORKREC.

4.1.8.2 Subroutine IES2_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES2 instrument configuration. The specific IES2 exceptions are listed in Table 2.

All inputs, processing, outputs, associated subroutines, and interfaces are as described for Subroutine IES_XCEPT (see 4.1.8.1).

4.1.8.3 Subroutine IES2A_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES2A instrument configuration. The specific IES2A exceptions are listed in Table 3.

All inputs, processing, outputs, associated subroutines, and interfaces are as described for Subroutine IES_XCEPT (see 4.1.8.1).

4.1.8.4 Subroutine IES3_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES3 instrument configuration. The specific IES3 exceptions will be listed in Table 4.

As of May 1994, the content and format of the IES3 telemetry data has not been defined to the level of detail needed to implement the exceptions processing for the instrument. This subroutine is currently implemented as a stub and will be fully implemented when the IES3 telemetry definition is finalized.

Table 2 IES2 exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
1	1	10	2	6/all	VIP Bias
2	1	12	2,6,10...	63/all	DM LLA
3	1	13	2	11/all	Vp EL
4	1	14	2	10/all	Ion Velocity
5	1	15	2	64/all	DM Signal Level
6	1	16	2	65/all	SM (WIBAN1) Range
7	1	17	2	66/all	DM (WIBAN2) Range
8	1	145	1	1/9	Cycle1 MSB
9	1	146	1	1/6-8	EP Mode
10	1	147	2	2/5	EP/RPA Flag
11	1	148	1	1/5	Test Mode
12	1	149	1	1/4	Bias Mode
13	1	150	1	1/3	Sweep Clock
14	1	151	1	1/2	PRF/Reset
15	1	152	1	2/9	RAM/PROM
16	1	153	1	2/8	Test/Flt
17	1	154	1	2/1-7	Prog. Vers.
18	1	155	2	6/7-8	VIP Set
19	1	156	2	6/1-5	VBias Volts
20	1	158	1&2	1/	Cycle Count
1	2	8	14,30,46...	62/all	DM Electron Temp.
2	2	9	1	6/all	Vap Monitor
3	2	11	1	7/all	RPA Monitor
4	2	12	4,8,12...	63/all	DM LLB
5	2	13	1	10/all	Vp RPA
6	2	20	14,30,46...	61/all	DSM Sensor Temp.
7	2	145	1	1/9	Cycle1 MSB
8	2	146	1	1/6-8	EP Mode
9	2	147	2	2/5	EP/RPA Flag
10	2	148	2	2/7	CkSum Err.
11	2	149	2	2/6	Uplink Flag
12	2	150			Null Value
13	2	151			Null Value
14	2	152	2	2/9	RAM Err.
15	2	153	2	2/8	Dump Flag
16	2	154	2	2/1-3	Serial Number
17	2	155	2	6/7-8	VIP Set
18	2	156	2	6/1-5	VBias Volts
19	2	158	1&2		Cycle Count

All inputs, processing, outputs, associated subroutines, and interfaces will be as described for Subroutine IES_XCEPT (see 4.1.8.1).

Table 3 IES2A exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
1	1	4	3,7,11..	5/all	TH+
2	1	5	3,7,11..	12/all	TO+
3	1	6	1,5,9...	12/all	TE+
4	1	10	2	6/all	VIP Bias
5	1	11	1,5,9...	11/all	EP Monitor
6	1	12	2,6,10...	63/all	DM LLA
7	1	13	2,6,10...	11/all	Vp EL
8	1	15	2	64/all	DM Signal Level
9	1	16	2	65/all	SM (WIBAN1) Range
10	1	17	2	66/all	DM (WIBAN2) Range
11	1	21	1,5,9...	5/all	RPA Thermistor
12	1	145	1	1/9	Cycle1 MSB
13	1	146	1	1/6-8	EP Mode
14	1	147	2	2/5	EP/RPA Flag
15	1	148	1	1/5	Test Mode
16	1	149	1	1/4	Bias Mode
17	1	150	1	1/3	Sweep Clock
18	1	151	1	1/2	PRF/Reset
19	1	152	1	2/9	RAM/PROM
20	1	153	1	2/8	Test/Flt
21	1	154	1	2/1-7	Prog. Vers.
22	1	155	2	6/7-8	VIP Set
23	1	156	2	6/1-5	VBias Volts
24	1	158	1&2	1/	Cycle Count
1	2	4	4,8,12...	5/all	NH+
2	2	5	4,8,12...	12/all	NO+
3	2	6	2,6,10...	12/all	NE+
4	2	8	14,30,46...	62/all	DM Electron Temp.
5	2	9	1	6/all	Vap Monitor
6	2	11	3,7,11...	11/all	RPA Monitor
7	2	12	4,8,12...	63/all	DM LLB
8	2	13	4,8,12...	11/all	Vp RPA
9	2	20	14,30,46...	61/all	DSM Sensor Temp.
10	2	21	2,6,10...	5/all	EP Thermistor
11	2	145	1	1/9	Cycle1 MSB
12	2	146	1	1/6-8	EP Mode
13	2	147	2	2/5	EP/RPA Flag
14	2	148	2	2/7	CkSum Err.
15	2	149	2	2/6	Uplink Flag
16	2	150			Null Value
17	2	151			Null Value
18	2	152	2	2/9	RAM Err.
19	2	153	2	2/8	Dump Flag
20	2	154	2	2/1-3	Serial Number
21	2	155	2	6/7-8	VIP Set
22	2	156	2	6/1-5	VBias Volts
23	2	158	1&2		Cycle Count

Table 4 IES3 exceptions list

Num	INTO Cycle	Word	FROM Cycle	Word/Bits	VARIABLE NAME
			T B S		

4.1.9 Subroutine STOREM.

a. Function.

Store the unpacked telemetry data from any of the IES instrument variants into the Common Format for the APGA program.

b. Inputs.

1. Argument list.

IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
KOUT	-	Pointer to the 1-minute data buffer to be reformatted

2. COMMON blocks.

/WORKREC/

IPHVAL	-	Unpacked Ephemeris data buffer
ITMUPK	-	Unpacked telemetry data buffer
NCYCLE	-	Telemetry data cycle indicators buffer
ITMSEC	-	Telemetry data times buffer
IXVAL	-	Telemetry data exceptions buffer

3. Files.

None.

c. Processing.

Storage of telemetry data from the original format (ITMUPK) into the Common Format (INOLS) is table driven. Each IES variant has its own unique storage mapping (table) to control the storage from its telemetry format and exceptions list into the Common Format.

The exceptions are accessed in order during storage into the output array; i.e.: the M^{th} occurrence of the exceptions flag (MEX) in the mapping table will cause the M^{th} value from the exceptions block to be stored in the current position of the output array.

A null flag (MNV) in the mapping table indicates that the data quantity for that Common Format location does not exist in the telemetry data for the given instrument type and/or data cycle. This flag causes a filler value (-1) to be stored in the corresponding position of the output array.

1. Store the Ephemeris data for the selected minute into the output buffer.

2. Establish a loop to store each second of the selected minute's Telemetry data into the output buffer, and, for each iteration of the loop:

   a. Store the Telemetry time value into the output buffer,

   b. Initialize the exceptions counter to zero,

   c. Determine the cycle type (1 or 2) for the current second,

   d. Establish a loop to store a value into each location of the output buffer (1 - 160).

      1. If the data cycle type is invalid, store a filler value (-1) into the current location of the output buffer.

      2. Otherwise, retrieve the mapping value for the current location from the proper instrument/cycle map and:

         a. Store the value from the telemetry data buffer location indicated by the mapping value into the current location of the output buffer.

         or

         b. When the mapping value is the exceptions flag (MEX), increment the exceptions counter and store the value from the exceptions buffer location indicated by the exceptions counter into the current location of the output buffer.

         or

         c. When the mapping value is the null value flag (MNV), store a filler value (-1) into the current location of the output buffer.

         or

         d. When the mapping value is the constants flag (MCO), store a constant value into the current location of the output buffer.

3. Return to PLIESPP.

d. Outputs.

1. Argument list.

   None.

2. COMMON blocks.

/OUTREC/

IPHVAL	-	One minute block of Ephemeris data
ISECOLS	-	One minute block of telemetry data times
INOLS	-	One minute block of reformatted telemetry data

e. Associated subroutines.

None.

f. Interfaces.

1. Called by PLIESPP.

2. Uses COMMON blocks WORKREC and OUTREC.

4.1.10 Subroutine OUTPUT.

a. Function.

Write the reformatted RIR, Ephemeris, Telemetry time, and Telemetry data buffers to the output file.

b. Inputs.

1. Argument list.

IESPREPFIL	-	Full path name for the Prepfile
IOUTUNIT	-	Unit number of the output file

2. COMMON blocks.

/OUTREC/

IPHOUT	-	Reformatted Ephemeris data buffer
ISECOLS	-	Telemetry data times buffer
INOLS	-	Reformatted Telemetry data buffer

/OUTRIR/

IRIROUT	-	Reformatted RIR data buffer

c. Processing.

1. If this is the first time through, open the output file and write the RIR data buffer.

2. Write the Ephemeris, Telemetry times, and Telemetry data buffers to the output file as a single record.

3. Return to PLIESPP.

d. Outputs.

1. Argument list.

None.

2. Files.

PREPFILE - The file to which the current 1-minute set of reformatted data is written.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by PLIESPP.

2. Uses COMMON blocks OUTREC and OUTRIR.

5 Environment

5.1 Equipment Environment.

The SSIES-2 Pre-processor (PLIESPP) has a major restriction arising from the structure of the raw data Mission Sensor file received from AFGWC and its preliminary processing at PL. This file contains data packed as four 9-bit telemetry words stored in 36-bit Sperry UNIVAC words, which are then embedded into 64-bit CONVEX (or 60-bit CYBER) words. The PLIESPP program reads the data into a VAX computer system and manipulates the data according to the specific constructs imposed on the input data by the VAX internal storage format. The PLIESPP program will require extensive modification to run properly on a system whose internal data storage conventions do not map exactly to those of the VAX.

The only other equipment constraints are that the hardware must have enough main memory to load and execute the PLIESPP program and have enough mass storage space for the various files (see Section 5.3 for file sizes).

5.2 Support Software.

The following system routines are required for the SSIES-2 pre-processor program PLIESPP:

1. VAX intrinsic functions: IBITS, MVBITS, BTEST

5.3 Data Base.

The data base file system consists of one input file and one output file, plus an optional diagnostic file and processing listings.

5.3.1 SSIES PHASE-I Data File.

a. Name: None designated; files are generally transferred from tape storage to disk before processing.

b. Usage: Generated at PL using data from AFGWC. This is the input file for the PLIESPP data unpacking and reformatting program.

c. Data Permanence: Archival tape storage is kept indefinitely.

d. Storage: This file is located on a tape storage, and is generally transferred to disk for processing. The record sizes are fixed, but the file sizes are variable. A nominal estimate of size for SSIES-2 is 160,000 words (about 1260 blocks) for 101 minutes of data (34 records of 4770 words each). This estimate approximates the amount of data in a single orbit.

e. Access type: SEQUENTIAL

f. Format: BINARY

g. Restrictions: The PLIESPP program uses the file on a read-only basis.

5.3.2 SSIES Common Format Data File.

a. Name: IESPREPFILE.

b. Usage: Produced by the PLIESPP program and used as the primary input to the APGA (IESCOMIN) program.

c. Data Permanence: Specified by PL/GPSP.

d. Storage: This file is located on a mass storage disk. The size of this file is variable, but a nominal estimate of size is 500,000 words (about 3900 blocks) for 101 fixed length records of 4888 words each. This estimate approximates the amount of data in a single orbit.

e. Access Type: SEQUENTIAL

f. Format: BINARY

g. Restrictions: None.

Appendix A: Time History Data Base

PHASE I SSIES-2 (F11 - F15) DATABASE FORMAT

There are 2544 CDC/NOS words (60 bits/word or 7.5 bytes) per physical record. Each record contains 3 minutes of data. For each minute there is ephemeris data and exactly 60 frames of telemetry data (one frame per second). Each minute of data requires 847 CDC/NOS words (60 bits/word). The three minutes of data are stored in words 1-2541. Word 2542 contains a code word to identify the spacecraft. The remaining two words are used to indicate the model magnetic field used. The last record of data for a day is followed by an End-of-File. If the last record for a day does not contain three minutes of SSIES data, the day number following the last good set of data is set to 999 and the remainder of the record is zero filled.

Should data be missing due to telemetry dropout or other reasons, zero fill is used at the end of the good data. The use of zero fill guarantees that all one minute groups are the same size. A 60-bit mapping word is used to indicate whether or not data exists for a particular second for the associated minute of data. If bit 60 is set to 1, the data for the zero second exists; if bit 59 is set to 1, the data for the next second exists, etc..

All angles are in degrees, except where indicated otherwise, and the altitude is in nautical miles. In the bit numbering sequence below: bit 60 is the most significant bit of a CDC word and bit 1 is the least significant bit.

CDC/NOS Word	Bits	Description
1	60-49	Geographic longitude (GLON) [×10]
	48-37	Geographic latitude (GLAT) [×10]
	36-31	Second (IS)
	30-25	Minute (IM)
	24-19	Hour (IH)
	18- 7	Day of year (JDAY)
	6- 1	Year (IYR) [Year=Year-1950]
2	60-49	Geomagnetic latitude at 110 km (RMLAT) [×10]
	48-37	Geomagnetic longitude at satellite (CMLONST) [×10]
	36-25	Geomagnetic latitude at satellite (GMLATST) [×10]
	24-13	Geographic Longitude at subsolar point (ALON) [×10]
	12- 1	Geographic latitude at subsolar paint (DEC) [×10]
3	60-49	Altitude at beginning of ephemeris minute (ALTBEG) - NMI
	48-37	Invariant latitude (RNVARLT) [×10]
	36-25	Geographic longitude of magnetic field line traced from the spacecraft to 110 km (CLON) [×10]
	24-13	Geographic latitude of magnetic field line traced from the spacecraft to 110 km (CLAT) [×10]
	12- 1	Geomagnetic longitude of magnetic field line traced from the spacecraft to 110 km (RMLON) [×10]
4	60-41	X coordinate of satellite unit position vector in ECI (XECOS) [$\times 10^5$]
	40-23	Magnetic local time of magnetic field line traced from the spacecraft to 110 km (RMLT) - SEC
	22-11	Altitude at end of ephemeris minute (ALTEND) - NMI
	10- 1	Filler
5	60-41	BX in 10ths of gamma
	40-21	Z coordinate of satellite unit position vector in ECI (ZECOS) [$\times 10^5$]
	20- 1	Y coordinate of satellite unit position vector in ECI (YECOS) [$\times 10^5$]

6	60-41	BZ in 10ths of gamma
	40-21	BY in 10ths of gamma
	20- 1	Orbital phase angle, in radians [$\times 10^5$]
7	60- 1	Mapping word (IMAP)
8-847		60 groups of 14 words (one group per second)
848-1694		Repeat order of words 1-847 for next minute
1695-2541		Repeat order of words 1-847 for next minute
2542	60- 1	Satellite ID (integer; 11 for satellite F11)
2543		Magnetic Field Model used stored in the 48 most significant bits of this word (6 bytes) as ASCII codes for text characters; the remaining 12 bits are vacant (zero fill).
2544		Vacant

NOTES:

1. The 14 CDC words containing the telemetry data actually consist of time (36 bits) and 84 9-bit data words. Each time word should be divided by 1024 to get the time in seconds. For F12-F15, only data words 1-80 are valid data.

2. For XECOS, YECOS and ZECOS, if the MSB of the 20-bit word is set to 1, the value is negative. To obtain the proper negative number subtract 1,048,575 from the value stored in the 20 bits.

3. For BX, BY and BZ, if the MSB of the 20-bit word is set to 1, the value is negative. Use the same procedure described in note 2. above to get the proper value.

4. For all latitude values, if the MSB of the 12-bit word is set to 1, the number is negative. To obtain the proper negative number, subtract 4095 from the value stored in the 12 bit word.

5. For the Magnetic Field model indicator (word 2543), character string information has been converted into a set of 8-bit bytes with each byte representing one ASCII character.

Appendix B: Common Format

RIR DATA BLOCK

Item #	Name (Type)	Units	Definition
1	IESTyp (INT)	N/A	SSIES instrument type code. 1: SSIES, 2: SSIES-2, 3: SSIES-2A, 4: SSIES-3.
2	MissId (INT)	N/A	4 digit Mission identifier. i.e.: 2546 is satellite F11.
3	RevNum (INT)	N/A	Revolution (orbit) number.
4	NumMin (INT)	N/A	Number of minutes of data in readout.
5	Nframe (INT)	N/A	1 second telemetry data frame count in readout.
6	NodYMD (INT)	N/A	Packed Nodal Year, Month, and Day. i.e.: YYMMDD.
7	Jday1 (INT)	N/A	Julian day at end of readout data.
8	Time1 (INT)	Seconds	Time of last readout data record.
9	Jday2 (INT)	N/A	Julian day at first readout data record.
10	Time2 (INT)	Seconds	Time of first readout data record.

EPHEMERIS DATA BLOCK

Item #	Name (Type)	Units	Definition
1	Lat1 (REAL)	Radians	Geodetic Latitude of Satellite subpoint at end of data minute.
2	Long1 (REAL)	Radians	Geodetic Longitude of Satellite subpoint at end of data minute.
3	Alt1 (INT)	Nmi	Satellite Altitude at end of data minute.
4	Jday1 (INT)	N/A	Julian Day at end of data minute.
5	Time1 (INT)	Seconds	Time of day at end of data minute.
6	Lat2 (REAL)	Radians	Geodetic Latitude of Satellite subpoint at start of data minute.
7	Long2 (REAL)	Radians	Geodetic Longitude of Satellite subpoint at start of data minute.
8	Alt2 (INT)	Nmi	Satellite Altitude at start of data minute.
9	Jday2 (INT)	N/A	Julian Day at start of data minute.
10	Time2 (INT)	Seconds	Time of day at start of data minute.
11	Xpos1 (REAL)	N/A	X unit orientation vector at end of data minute.
12	Ypos1 (REAL)	N/A	Y unit orientation vector at end of data minute.
13	Zpos1 (REAL)	N/A	Z unit orientation vector at end of data minute.
14	Xpos2 (REAL)	N/A	X unit orientation vector at start of data minute.
15	Ypos2 (REAL)	N/A	Y unit orientation vector at start of data minute.
16	Zpos2 (REAL)	N/A	Z unit orientation vector at start of data minute.
17	Lat1 (INT)	Radians x 10000	Geodetic Latitude of Satellite subpoint at end of data minute.
18	Long1 (INT)	Radians x 10000	Geodetic Longitude of Satellite subpoint at end of data minute.
19	Alt1 (REAL)	Nmi	Satellite Altitude at end of data minute.
20	Lat2 (INT)	Radians x 10000	Geodetic Latitude of Satellite subpoint at start of data minute.
21	Long2 (INT)	Radians x 10000	Geodetic Longitude of Satellite subpoint at start of data minute.
22	Alt2 (REAL)	Nmi	Satellite Altitude at start of data minute.

23	Phi1 (REAL)	Radians	Angle on the orbital plane between the ascending node and the satellite location at end of data minute.
24	Phi2 (REAL)	Radians	Angle on the orbital plane between the ascending node and the satellite location at start of data minute.
25	Nrev (INT)	N/A	Revolution (Orbit) Number.
26	R^+ Number (INT)	N/A	Readout Rev number relative to start of satellite day.
27	Filler	N/A	Unused data location.
28	Filler	N/A	Unused data location. Note that the APGA program stores the packed Nodal Year, Month, Day value from the RIR record into this word of the Ephemeris block.

DATA FRAME TIME BLOCK

Item #	Name (Type)	Units	Definition
1 - 60	Ifrmtim (INT)	Seconds	Data frame validity time in seconds since midnight.

INSTRUMENT DATA FRAME BLOCK

Cycle 1

Item #	Name (Type)	Units	Definition
1	Cycle1ID (INT)	N/A	Cycle counter
2	Config1ID (INT)	N/A	Configuration ID
3	OLSCmdMon (INT)	N/A	OLS command monitor
4	MPTemp:H+ (INT)	N/A	Microprocessor H+ temperature
5	MPTemp:O+ (INT)	N/A	Microprocessor O+ temperature
6	MPTemp:e (INT)	N/A	Microprocessor electron temperature
7	CurrentMon (INT)	N/A	Current monitor
8	ElmeterTemp (INT)	N/A	Electrometer temperature
9	VAperMon (INT)	N/A	Aperture voltage monitor
10	BiasMon (INT)	N/A	Bias voltage monitor
11	ElectronMon (INT)	N/A	Microprocessor EP flags
12	DMLLA (INT)	N/A	Driftmeter LLA
13	MPEPVehPot (INT)	N/A	Microprocessor EP vehicle potential
14	MPRamVel (INT)	N/A	Microprocessor ion velocity
15	DMSigLev (INT)	N/A	DM signal level
16	SMWIBAN1 (INT)	N/A	SM WIBAN1 range
17	DMWIBAN2 (INT)	N/A	DM WIBAN2 range
18	Spare		Unused
19	Spare		Unused
20	Spare		Unused
21	RPATherm (INT)	N/A	RPA thermistor
22-45	DSM:EL/AMP (INT)	N/A	DSM ELE/AMP (24 values)
46-57	DSM:Drift (INT)	N/A	DSM drift (12 values)

58-93	RPACurrent (INT)	N/A	RPA current (36 values)
94-117	EPCurrent (INT)	N/A	EP current (24 values)
118-126	SMFIBA (INT)	N/A	SM filter (9 values)
127-132	DMFIBA (INT)	N/A	DM FIBA (6 values)
133-138	EPSweepMon (INT)	N/A	EP sweep monitor (6 values)
139-144	RPASweepMon (INT)	N/A	RPA sweep monitor (6 values)
145	Cycle1MSB (INT)	N/A	Cycle 1 MSB
146	EPMode (INT)	N/A	EP mode flag (3 bits)
147	EP/RPAFlag (INT)	N/A	EP/RPA telemetry flag bit
148	TestMode (INT)	N/A	Test mode flag bit
149	BiasMode (INT)	N/A	Bias mode flag bit
150	SweepClock (INT)	N/A	Sweep clock flag bit
151	PRFReset (INT)	N/A	Program Reset Flag bit
152	RAM/PROM (INT)	N/A	Program upload flag
153	Test/Flt (INT)	N/A	Test/flight flag
154	ProgVer (INT)	N/A	Program version (7 bits)
155	VIPSet (INT)	N/A	VIP setting flag (2 bits)
156	VBIAS (INT)	N/A	VBIAS voltage (5 bits)
157	InstCode (INT)	N/A	Instrument code
158	CycleCount (INT)	N/A	Calculated cycle count
159-160	Spare		Unused (2 values)

INSTRUMENT DATA FRAME BLOCK (continued)

Cycle 2

Item #	Name (Type)	Units	Definition
1	Cycle2ID (INT)	N/A	Cycle counter
2	Config2ID (INT)	N/A	Configuration ID
3	DSMCmdMon (INT)	N/A	DSM command monitor
4	MPDens:H+ (INT)	N/A	Microprocessor H+ density
5	MPDens:O+ (INT)	N/A	Microprocessor O+ density
6	MPDens:e (INT)	N/A	Microprocessor electron density
7	ADCTempMon (INT)	N/A	ADC temperature
8	DMTempElec (INT)	N/A	DM electron temperature
9	VAperMon (INT)	N/A	Aperture voltage monitor
10	BiasMon (INT)	N/A	Bias voltage monitor
11	RPAIonMon (INT)	N/A	Microprocessor RPA flags
12	DMLLB (INT)	N/A	Driftmeter LLB
13	MPRPAVehPot (INT)	N/A	Microprocessor RPA vehicle potential
14	MPRamVel (INT)	N/A	Microprocessor ion velocity
15	DMSigLev (INT)	N/A	DM signal level
16	SMWIBAN1 (INT)	N/A	SM WIBAN1 range
17	DMWIBAN2 (INT)	N/A	DM WIBAN2 range
18	Subcom1 (INT)	N/A	Subcom 1
19	Subcom2 (INT)	N/A	Subcom 2
20	DSMSensTemp (INT)	N/A	DM sensor temperature
21	EPTherm (INT)	N/A	EP thermistor
22-45	DSM:EL/AMP (INT)	N/A	DSM ELE/AMP (24 values)

46-57	DSM:Drift (INT)	N/A	DSM drift (12 values)
58-93	RPACurrent (INT)	N/A	RPA current (36 values)
94-117	EPCurrent (INT)	N/A	EP current (24 values)
118-126	SMFIBA (INT)	N/A	SM filter (9 values)
127-132	Spare		Unused (6 values)
133-138	EPSweepMon (INT)	N/A	EP sweep monitor (6 values)
139-144	RPASweepMon (INT)	N/A	RPA sweep monitor (6 values)
145	Cyc1MSB (INT)	N/A	Cycle 1 MSB
146	EPMode (INT)	N/A	EP mode flag (3 bits)
147	EP/RPAFlag (INT)	N/A	EP/RPA telemetry flag bit
148	Checksum (INT)	N/A	Checksum error flag bit
149	UplinkFlag (INT)	N/A	Uplink flag bit
150	Spare		Unused
151	VBiasMon (INT)	N/A	VBIAS monitor flag bit
152	RAMError (INT)	N/A	RAM error flag bit
153	DumpFlag (INT)	N/A	Dump flag
154	SerialNum (INT)	N/A	Serial number (3 bits)
155	VIPSet (INT)	N/A	VIP setting flag (2 bits)
156	VBIAS (INT)	N/A	VBIAS voltage (5 bits)
157	InstCode (INT)	N/A	Instrument code
158	CycleCount (INT)	N/A	Calculated cycle count
159-160	Spare		Unused (2 values)

Word Number	Word Identification											
1 - 12	Cycle 1 ID	Config 1 ID	OLS Cmd Mon	MP Temp:H+	MP Temp:O+	MP Temp:e	Current Monitor	Elmeter Temp	Vaper Monitor	Bias Monitor	Electron Monitor	DM LLA
	Cycle 2 ID	Config 2 ID	DSM Cmd Mon	MP Dens:H+	MP Dens:O+	MP Dens:e	ADC Temp Monitor	DM Temp Electron			RPA Ion Monitor	DM LLB
13 - 24	MP EP Veh Pot	MP Ram Velocity	DM Signal Level	SM WIBAN1 Range	DM WIBAN2 Range	Not Used	Not Used	Not Used	RPA Therm	DSM: EL/AMP	-->	-->
	MP RPA Veh Pot					Subcom 1	Subcom 2	DSM Sens Temp	EP Therm			
25 - 36	DSM: EL/AMP	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->
37 - 48	DSM: ELE/AMP	-->	-->	-->	-->	-->	-->	-->	-->	DSM: DRIFT	-->	-->
49 - 60	DSM: DRIFT	-->	-->	-->	-->	-->	-->	-->	-->	RPA Current	-->	-->
61 - 72	RPA Current	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->
73 - 84	RPA Current	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->

Table B-1A IESPREPFILE Common Format and Identification (part 1)

Word Number	Word Identification											
85 - 96	RPA Current	-->	-->	-->	-->	-->	-->	-->	-->	EP Current	-->	-->
97 - 108	EP Current	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->
109 - 120	EP Current	-->	-->	-->	-->	-->	-->	-->	-->	SMFIBA1	SMFIBA2	SMFIBA3
121 - 132	SMFIBA4	SMFIBA5	SMFIBA6	SMFIBA7	SMFIBA8	SMFIBA9	DMFIBA1 Not Used	DMIFBA2 Not Used	DMFIBA3 Not Used	DMFIBA4 Not Used	DMFIBA5 Not Used	DMFIBA6 Not Used
133 - 144	EP Sweep Monitor	-->	-->	-->	-->	-->	RPA Sweep Monitor	-->	-->	-->	-->	-->
145 - 156	Cycle 1 MSB	EP Mode	EP/RPA Flag	Test Mode Checksum	Bias Mode Uplink Flag	Sweep Clock Not Used	PRF Reset VBias Monitor	RAM/PROM Op. RAM Error	Test/Flt Flag Dump Flag	Prog. Version Serial Number	VIP Setting	VBias Volts
157 - 160	Inst. Code	Cycle Count	Not Used	Not Used								

All cycles — Identification of telemetry item appearing in both even and odd cycles

Odd Cycles — Identification of telemetry item appearing in odd cycles

Even Cycles — Identification of telemetry item appearing in even cycles

Not Used — Spare word. Contains -1.

Table B-1B IESPREPFILE Common Format and Identification (part 2)

Notes:

1. When a particular SSIES telemetry variant does not contain data for a common format word allocation, the preprocessor program fills that word with a (-1), i.e.: only SSIES-3 has 36 words of RPA current data, so, for SSIES, SSIES-2, and SSIES-2A, words 82 - 93 will contain (-1).

2. The sequence of values appearing in the two subcommutator words of the SSIES-2 and SSIES-2A telemetry are identified in the following table.

Cycle Second	Subcom 1	Subcom 2
2,18,34,50,66,...	REG3A	REG3B
4,20,36,52,68,...	REG3C	REG3D
6,22,38,54,70,...	REG2A	REG2B
8,24,40,56,72,...	REG2C	REG2D
10,26,42,58,74,...	REG1A	REG1B
12,28,44,60,76,...	REG1C	REG1D
14,30,46,62,78,...	SENSTEMP	ELECTEMP
16,32,48,64,80,...	COMDATD	RELAYFLG

Table B-2 Subcommutator Format

The following table displays the values of the SSIES-2 and SSIES-2A subcom command registers, and the associated commands which set the registers. For each register, a reported zero value is derived from a nine-bit telemetry zero value, while a reported one value is derived from a nine-bit telemetry 511 value. The threshold between reported zero or one values is considered to be a nine-bit telemetry 256 value. The registers are latching registers, so that the instrument status can be determined from the subcom even after the associated commands have been superseded in the telemetry DSM command monitor. However, acquiring the appropriate values for one register set (REG1, REG2, or REG3) from the subcom requires at least three seconds (two successive even cycles), and acquiring the entire subcom set requires sixteen seconds.

Command	REG1D	REG1C	REG1B	REG1A	Mnemonic	Command	REG2D	REG2C	REG2B	REG2A	Mnemonic	Command	REG3D	REG3C	REG3B	REG3A	Mnemonic
82	0	0	0	0	WB2AUTO	88	0	0	0	0	WB1AUTO	90	0	0	0	0	DREP00
	0	0	0	1	spare		0	0	0	1	spare		0	0	0	1	spare
	0	0	1	0	spare		0	0	1	0	spare	92	0	0	1	0	DREP10
	0	0	1	1	spare		0	0	1	1	spare	93	0	0	1	1	DREP15
	0	1	0	0	spare		0	1	0	0	spare	94	0	1	0	0	DREP20
	0	1	0	1	spare		0	1	0	1	spare	95	0	1	0	1	DREP25
	0	1	1	0	spare		0	1	1	0	spare	96	0	1	1	0	DREP30
	0	1	1	1	spare		0	1	1	1	spare		0	1	1	1	spare
83	1	0	0	0	WB2RNG1	89	1	0	0	0	WB1RNG1	98	1	0	0	0	0WIGLO
84	1	0	0	1	WB2RNG2	8A	1	0	0	1	WB1RNG2	99	1	0	0	1	0WIGHI
85	1	0	1	0	WB2RNG3	8B	1	0	1	0	WB1RNG3	9A	1	0	1	0	1WIGLO
86	1	0	1	1	WB2RNG4	8C	1	0	1	1	WB1RNG4	9B	1	0	1	1	1WIGHI
87	1	1	0	0	WB2RNG5	8D	1	1	0	0	WB1RNG5	9C	1	1	0	0	2WIGLO
	1	1	0	1	spare	16,19	1	1	0	1	SENPOT	9D	1	1	0	1	2WIGHI
	1	1	1	0	spare	30-4C	1	1	1	0	VBIAS	9E	1	1	1	0	3WIGLO
	1	1	1	1	spare		1	1	1	1	spare	9F	1	1	1	1	3WIGHI

Table B-3 Subcom Register Definitions

Appendix C: Reference Materials

Glossary

AFGWC	-	Air Force Global Weather Central
APGA	-	SSIES analysis program at SFC (designated IESCOMIN at PL)
BNBA	-	SSIES data re-formatter at SFC
Common Format	-	SSIES data format generated by both BNBA and PLIESPP for all SSIES versions, for use by APGA
DM	-	Driftmeter (one of the SSIES components)
DSM	-	Driftmeter/Scintillation meter
DMSP	-	Defense Meteorological Satellite Program
EP	-	Electron probe (one of the SSIES components)
IES	-	Identifier for SSIES (original) format
IES2	-	Identifier for SSIES-2 format
IES2A	-	Identifier for SSIES-2A format
IES3	-	Identifier for SSIES-3 format
IESCOMIN	-	PL designation for SFC SSIES analysis program
IESPREPFILE	-	Common Format file used for processing by APGA
MSB	-	Most significant bit
OLS	-	Operational line scanner (DMSP instrument)
PHASE-I	-	SSIES pre-processing program at PL for generating DMSP THDB
PL	-	Phillips Laboratory
PL/GP	-	Phillips Laboratory/Geophysics Directorate
PLIESPP	-	SSIES data re-formatter at PL
RIR	-	Readout Information Record, for SSIES data
RPA	-	Retarding potential analyzer (one of the SSIES components)
SFC	-	Air Force Space Forecast Center
SM	-	Scintillation meter (one of the SSIES components)
SSIES	-	Special Sensor for Ions, Electrons, and Scintillation (DMSP spacecraft F8 - F10)
SSIES-2	-	Special Sensor for Ions, Electrons, and Scintillation (DMSP spacecraft F11 - F15)
SSIES-2A	-	Special Sensor for Ions, Electrons, and Scintillation (DMSP spacecraft F11 - F15, with 80 telemetry words/second instead of 84 telemetry words/second)
SSIES-3	-	Special Sensor for Ions, Electrons, and Scintillation (DMSP spacecraft F16 - F20)
THDB	-	Time History Data Base

Documents

1. Delorey, D.E., Pruneau, P.N., Parsons, C.M., and Donovan, B.J., Database Development for the DMSP Experiments, GL-TR-90-0028, 31 January 1990, ADA224346.

2. Rich, F.J., User's Guide for the Topside Ionospheric Plasma Monitor (SSIES, SSIES-2, SSIES-3) on Spacecraft of the Defense Meteorological Satellite Program (DMSP): Volume 1, Technical Description, PL-TR-94-2187.

3. Cornelius, J.R., and Mazzella, A.J., User's Guide for the Topside Ionospheric Plasma Monitor (SSIES, SSIES-2, SSIES-3) on Spacecraft of the Defense Meteorological Satellite Program (DMSP): Volume 2, Programmer's Guide for Software at AFSFC, PL-TR-94-2270.

4. Cornelius, J.R., and Mazzella, A.J., User's Guide for the Topside Ionospheric Plasma Monitor (SSIES, SSIES-2, SSIES-3) on Spacecraft of the Defense Meteorological Satellite Program (DMSP): Volume 3, Program Maintenance Manual, PL-TR-94-2271.